MANUAL DE USUARIO
Guía de Entregas con Subversion de proyectos de movilidad

Versión 1.0

Área de Arquitecturas

Agencia de Informática y Comunicaciones de la Comunidad de Madrid

Hoja de Control

Título	Guia de entregas con subversion de proyectos de movilidad		
Documento de Referencia			
Responsable	Área de Arquitecturas		
Versión	1.0	**Fecha Versión**	25/09/2014

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Área de Arquitecturas	02/06/2011

Índice

1. INTRODUCCIÓN 4

2. ESTRUCTURA DE REPOSITORIOS DE SUBVERSION EN ICM 4

3. ESTRUCTURA DE UN PROYECTO DE APP MOVIL 5

3.1. ESTRUCTURA DEL DIRECTORIO PROYECTO DE CADA PLATAFORMA 7

3.1.1. Android 7

3.1.2. IOS 9

3.1.3. Window Phone 8 9

4. REALIZACIÓN DE UNA ENTREGA PARA SU INSTALACIÓN EN LOS ENTORNOS DE ICM 12

1. INTRODUCCIÓN

En este documento se explica el uso de la herramienta de control de versiones Subversion para la entrega de proyectos de movilidad para ICM.

2. ESTRUCTURA DE REPOSITORIOS DE SUBVERSION EN ICM

Todo proyecto en ICM tiene un repositorio de Subversion propio. Se puede acceder al repositorio de un determinado proyecto utilizando el cliente de Subversion incluido con Eclipse (Subclipse), con cualquier otro client de subversion o también utilizando un browser y accediendo a la URL del repositorio (en este caso se accede en modo solo lectura)

URL del servidor de Subversion de un proyecto

URL:
https://**RutaServidorSVN**/svn/**ProyectoMinusculas**/

Valores de las variables:

- **RutaServidorSVN**: Ruta del servidor de Subversion (por defecto es https://subversion01.madrid.org:8443)
- **ProyectoMinusculas**: Nombre del proyecto en minúsculas (Ej: bsta).

Ejemplo:
https://subversion01.madrid.org:8443/svn/cisa/

Si accedemos con un explorador a la URL de un repositorio podemos navegar por su contenido:

Nota
Cada repositorio de un proyecto tiene asignada una **seguridad** para que sólo las personas autorizadas puedan ver el código y realizar modificaciones sobre él.

Todo repositorio de un proyecto en ICM tiene la misma estructura, que consiste en las siguientes carpetas:

- **entregas**: Cada vez que se desea instalar un proyecto o módulo en cualquier entorno de ICM, se crea una "foto" del código en un momento determinado, que será la que se instale en el entorno deseado. En la nomenclatura de Subversion, una foto congelada del código se denomina **TAG**. Todos los tags creados para instalar en los distintos entornos de ICM se crean en esta carpeta "entregas".
- **instalaciones**: Se trata de una carpeta de uso interno por las unidades de Paso a Producción. (Esta carpeta solo la ve el grupo indicado)
- **vXXXX**: Cada versión de desarrollo que se mantenga para el proyecto tendrá una carpeta específica. Por ejemplo, si se está desarrollando una nueva versión del proyecto y a la vez se está realizando mantenimiento correctivo sobre el actual, habrá dos carpetas, una para cada versión.

3. ESTRUCTURA DE UN PROYECTO DE APP MOVIL

Los proyectos de la Comunidad de Madrid se organizan como módulos técnicos. En el caso de aplicaciones moviles va a existir un módulo técnico para cada plataforma en la que se despliegue la aplicación, con la siguiente nomenclatura:

- **XXXX_IOS_NombreApp**, para tecnología IOS.
- **XXXX_ANDROID_NombreApp**, para tecnología ANDROID.
- **XXXX_WPHONE_NombreApp**, para tecnología WINDOWS PHONE.

donde:

- "XXXX" es el código de la Aplicación registrada en POAPS.
- "*NombreApp*" es el nombre de la Aplicación para dispositivos móviles.

Por ejemplo:

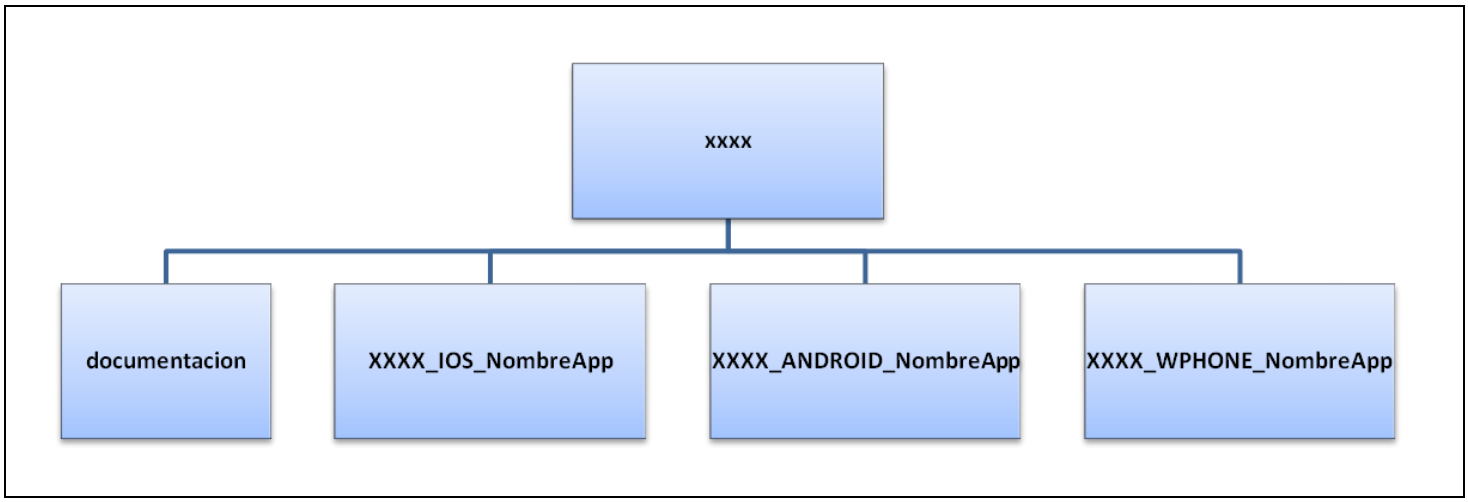
- **CISA_IOS_CITASANITARIA**, para tecnología IOS.
- **CISA_ANDROID_CITASANITARIA**, para tecnología ANDROID.
- **CISA_WPHONE_CITASANITARIA**, para tecnología WINDOWS PHONE.

Todos los módulos técnicos de un proyecto se incluyen dentro de una carpeta raiz llamada 'XXXX', (CODIGO POAPS)

Bajo el directorio raíz, colgarán los siguientes directorios:

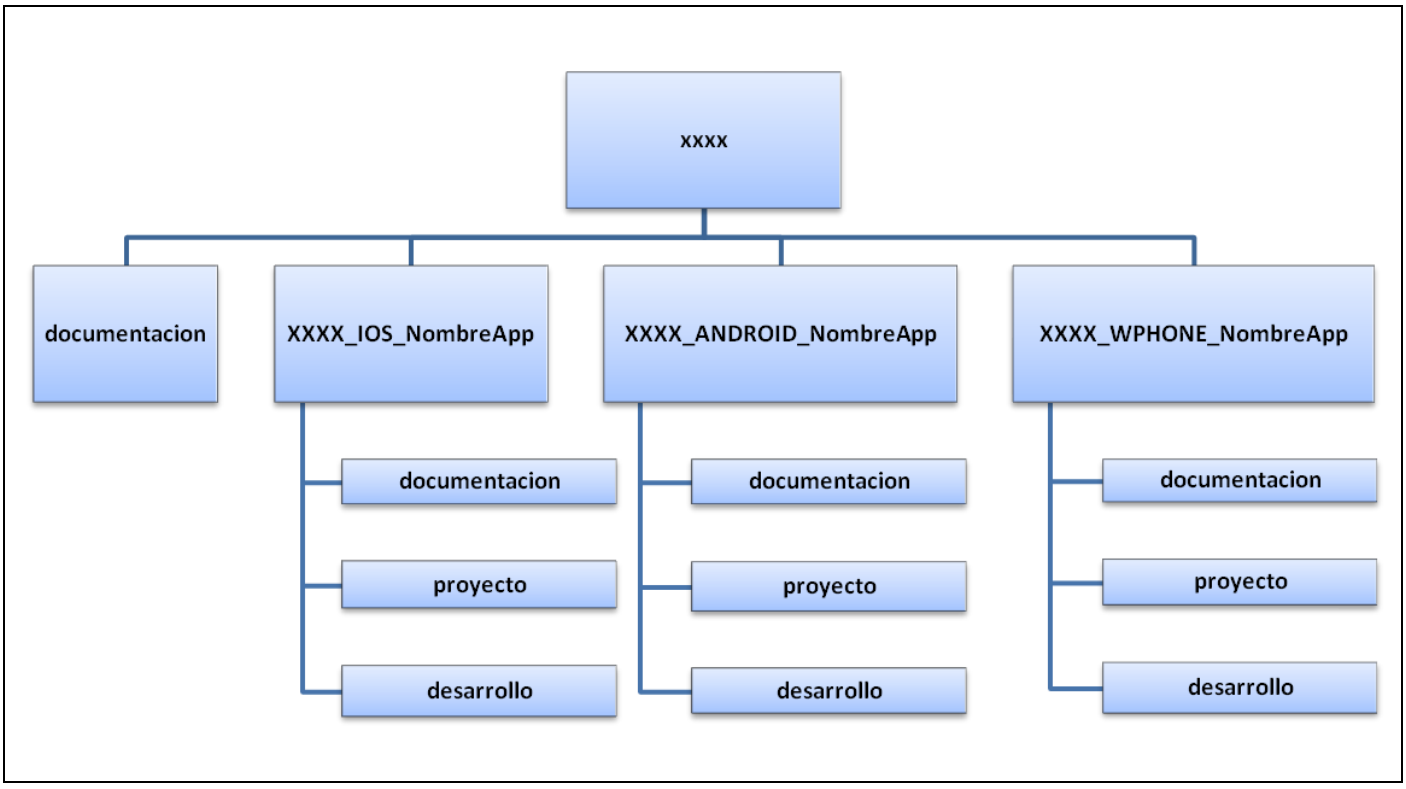
- Documentacion: Directorio que contiene la documentación general de la aplicación y común a todas las plataformas en las que se realiza la entrega: Análisis Funcional (DDF), Diseño Técnico (DDT), Documento de Pruebas (DPS), etc.
- Un directorio por cada una de las tecnologías móviles para las que se realiza la entrega: "IOS", "ANDROID"

y/o “WPHONE” con la nomenclatura indicada para cada uno de los módulos técnicos.



Bajo la carpeta correspondiente a cada tecnología colgarán los siguientes directorios:

- Documentacion: Directorio que contiene la documentación propia de la tecnología. Entre estos documentos debe existir la **Ficha de Publicación** en el Store correspondiente y las capturas de imágenes necesarias para la publicación y resto de recursos referenciados en esa ficha.
- **Proyecto:** Directorio que contiene los fuentes del proyecto en la plataforma correspondiente
- **Desarrollo**. Directorio que contiene el paquete entregable para las pruebas de validación de la APP (.APK para Android o .IPA para IOS)



3.1. Estructura del directorio PROYECTO de cada Plataforma

El directorio PROYECTO deberá tener la siguiente estructura según la plataforma:

3.1.1. Android

La estructura de directorios por debajo de la carpeta PROYECTO de un módulo técnico de tipo ANDROID deberá ser la siguiente:

- **SRC**. Directorio que contiene las clases java del proyecto.
- **GEN**. Directorio que contiene las clases java autogeneradas por el SDK para el entorno de desarrollo. Hacen referencia a los recursos definidos de la carpeta RES.
- **ASSETS**. Directorio que contiene recursos del proyecto que son accesibles en tiempo de ejecución mediante el AssetManager. Al ser accedidos son tratados como un stream de bytes. La estructura interna de este directorio es libre.
  Los fuentes **HTML5**, en caso de que la aplicación haga uso de ellos, colgarán de este directorio en un directorio de nombre WWW.
- **RES**. Directorio que contiene recursos del proyecto que son accesibles en tiempo de ejecución mediante la clase R generada por el SDK. Al ser accedidos son tratados en su formato original, sin tratamiento previo. La estructura interna de este directorio está definida y puede contener los siguientes directorios:
  - DRAWABLE. Directorio que contiene las imágenes y los descriptores xml de imágenes que utiliza la aplicación.
  - LAYOUT. Directorio que contiene los ficheros xml que configuran las diferentes pantallas de la interfaz de usuario de la aplicación.
  - MENU. Directorio que contiene los ficheros xml de menú de la aplicación.
  - VALUES. Directorio que contiene ficheros xml de configuración de variables con valores como cadenas de caracteres (*strings*), tamaños (*dimens*) o colores (*colors*).
  - ANIM. Directorio que contiene ficheros xml de animaciones Tween para realizar transformaciones simples en los contenidos de un objeto View.
  - ANIMATOR. Directorio que contiene ficheros XML con animaciones de propiedades.
  - XML. Directorio que contiene otros ficheros XML de la aplicación tales como config.xml.
  - RAW. Otros recursos adicionales que no se encuentran en formato xml.
- **LIBS**. Librerías JAR que utiliza la aplicación.
- **CORDOVA**. Librerías y utilidades phonegap (cordova) utilizadas para la generación/modificación del proyecto Android en base a fuentes HTML5.

- **CORDOVALIB**. Proyecto Android Cordova incluido en la aplicación que permite a la aplicación, mediante la inclusión y uso de sus clases, utilizar funcionalidades nativas del dispositivo.

- **BIN**. Clases compiladas de la aplicación.

En el raiz de la carpeta PROYECTO se ubicarán los siguientes ficheros:

- **AndoridManifest.xml**. Fichero descriptor de la aplicación:
  - Nombre del paquete java como Identificador único de la aplicación.
  - Nombre de la aplicación.
  - Versión de la aplicación.
  - Icono de la aplicación.
  - Orientación de la pantalla.
  - Descripción de los componendes de la aplicación.
  - Permisos de la aplicación con respecto al API.
  - Permisos de la aplicación respecto a otras aplicaciones.
  - Permisos de otras aplicaciones respecto a la aplicación.
  - Versión mínima requerida del API de Android sobre la que corre la aplicación.
  - Versión del API de Android con el que se ha compilado la aplicación.
  - Lista las librerías referenciadas por la aplicación.

- **Project.properties**. Fichero de configuración del proyecto.

- **.classpath**. Fichero de configuración del classpath del proyecto.

- **.project**. Fichero de configuración del proyecto.

Ejemplo de entrega en la carpeta ANDROID:

3.1.2. IOS

La estructura de directorios y ficheros será la siguiente:

- **XXXX.XCODEPROJ**. Directorio, donde XXXX es el nombre del proyecto, que contiene ficheros de configuración interna del proyecto Xcode.
  XXXX es el nombre del proyecto de Xcode.
- **XXXX**. Directorio, donde XXXX es el nombre del proyecto, que contiene los recursos del proyecto.
- **CORDOVA**. Librerías y utilidades phonegap (cordova) utilizadas para la generación/modificación del proyecto Xcode en base a fuentes HTML5.
- **CORDOVALIB**. Proyecto Xcode Cordova incluido en la aplicación que permite a la aplicación, mediante la inclusión y uso de sus clases, utilizar funcionalidades nativas del dispositivo.
- **WWW**. Directorio que contiene los fuentes HTML5 en caso de que la aplicación haga uso de ellos.

Ejemplo de entrega en la carpeta IOS:

3.1.3. Window Phone 8

La estructura de directorios y ficheros de la carpeta PROYECTO de un módulo técnico de tipo Windows Phone será la siguiente:

- **PROPERTIES**. Directorio que contiene los siguientes ficheros de configuración:

  - AppManifest.xml. Fichero manifesto Silverlight descriptor general de una aplicación.
  - AssemblyInfo.cs. Fichero de configuración con meta-datos propios de la aplicación, como el título, la descripción, la compañía, número de versión, …
  - WMAppManifest.xml. Fichero manifesto descriptor de la aplicación Windows Phone que describe aspectos de la aplicación como el título, ubicación de iconos, permisos de la aplicación, …

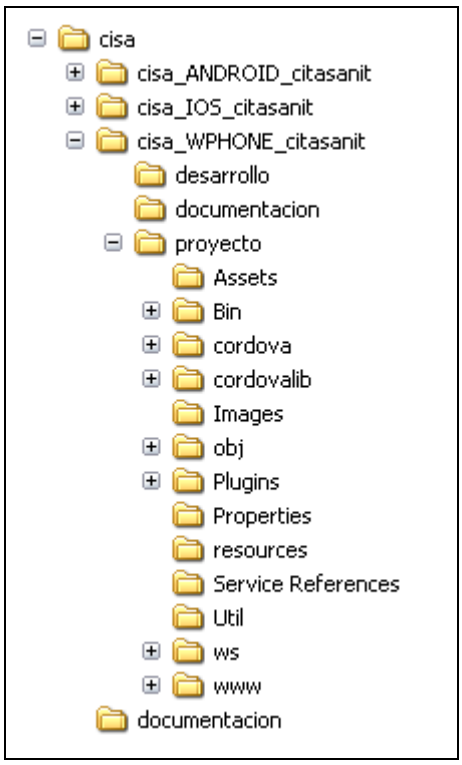
- **RESOURCES**. Directorio que recursos del proyecto, audio, vídeo, ...
- **ASSETS**. Directorio que contiene recursos del proyecto que son accesibles en tiempo de ejecución de la aplicación.
  La estructura interna de este directorio es libre.
- **IMAGES**. Directorio que contiene imágenes de la aplicación nativa Windows Phone.
- **REFERENCES**. Contiene las referencias a librerías Windows Phone.
- **SERVICE REFERENCES**. Contiene las referencias a servicios alojados en servidores determinados.
- **CORDOVA**. Librerías y utilidades phonegap (cordova) utilizadas para la generación/modificación del proyecto Windows Phone en base a fuentes HTML5.
- **CORDOVALIB**. Clases cs de la librería Cordova que permiten realizar debug en el proyecto.
- **PLUGINS**. Clases cs Cordova que permiten, a los fuentes HTML5, utilizar funcionalidades nativas del dispositivo.
- **BIN**. Clases compiladas de la aplicación.
- **WWW**. Directorio que contiene los fuentes HTML5 en caso de que la aplicación haga uso de ellos.

En la propia carpeta PROYECTOS se ubicarán los siguientes ficheros:

- APP.XAML. Fichero descriptor de la aplicación en el que se describen datos globales y de configuración de la aplicación.
- MAINPAGE.XAML. Página inicial de una aplicación Windows Phone nativa.
- APPLICATIONICON.PNG. Icono que representa la aplicación en los menús del teléfono.
- BACKGROUND.PNG. Imagen que se muestra al usuario al inicio de la aplicación nativa mientras se está cargando ésta.
- SPLASHSCREENIMAGE.JPG. Imagen que se muestra al usuario al inicio de la aplicación Phonegap/Cordova mientras se está cargando ésta.

- CONFIG.XML. Fichero de configuración de aplicación phonegap/cordova/HTML5.
- VERSION. Fichero de versión phonegap, en caso de utilizarse Phonegap/Cordova.

Ejemplo de entrega en la carpeta WINDOWS_PHONE_8:



4. REALIZACIÓN DE UNA ENTREGA PARA SU INSTALACIÓN EN LOS ENTORNOS DE ICM

Para poder pasar una versión de nuestro proyecto a cualquiera de los entornos de ICM (validación, producción, etc.) es necesario realizar una "foto" del código en un momento determinado, que será la que se instale en el entorno deseado. En la nomenclatura de Subversion, una foto congelada del código se denomina **TAG**. Por tanto, para pasar una versión a cualquier entorno de ICM se ha de crear un TAG, en la carpeta "**entregas**" del repositorio del proyecto.

URL del servidor de Subversion para crear un TAG

URL:
https://**RutaServidorSVN**/svn/**ProyectoMinusculas**/entregas/**Fecha_[NOMBREMODULO]_Version**

Valores de las variables:

- **RutaServidorSVN**: Ruta del servidor de Subversion (por defecto es https://subversion01.madrid.org:8443)
- **ProyectoMinusculas:** Nombre del proyecto en minúsculas (Ej: cisa).
- ***[NOMBREMODULO] (Opcional)***: Nombre del módulo instalado EN MAYÚSCULAS. Este parámetro **es opcional**, y puede incluirse si se desea realizar una entrega de un solo módulo en concreto, y quiere distinguirse de otro tipo de entregas totales. Si el nombre del módulo fuese muy largo, puede utilizarse una **abreviatura**.
- **Fecha:** Fecha en la que se crea el TAG, con formato AAAAMMDD
- **Version:** Versión de la que se trata. Ej: v1.0.3

Ejemplos:
https://subversion01.madrid.org:8443/svn/cisa/entregas/20110602_v1.0.3
https://subversion01.madrid.org:8443/svn/cisa/entregas/20110602_CISA_ANDROID_CITASANITARIA_v1.0.3

Dentro de la nueva carpeta en entregas irá el contenido del proyecto según la estructura indicada en el apartado anterior.